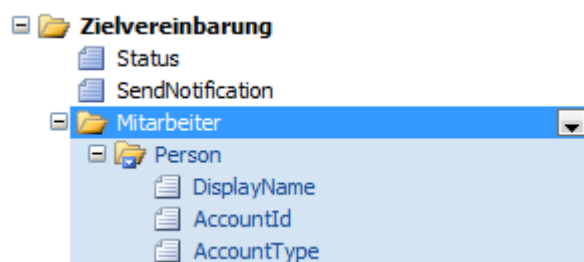


## InfoPath, Contact Selector & MOSS User Profile Webservice

Während der Arbeit an einer InfoPath-Form stand ich neulich vor folgender Aufgabe: Eine von mehreren im Formular erwähnten und über Contact Selector auswählbaren Personen ist - mit hoher Wahrscheinlichkeit - der Mitarbeiter, der die Erfassung von Informationen initiiert. Es wäre also vorteilhaft, wenn beim ersten Öffnen des Formulars die Kontaktdaten des Mitarbeiters automatisch ausgefüllt wären.

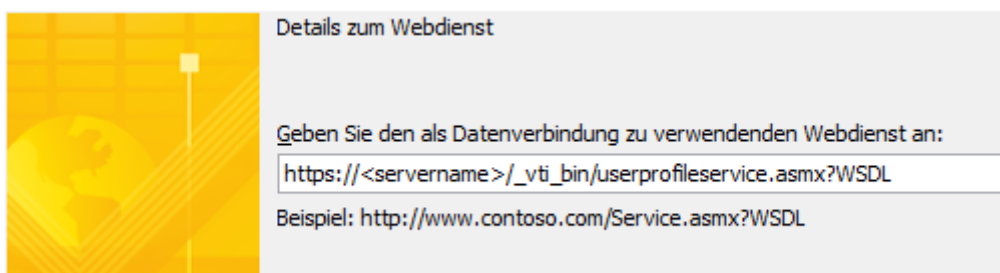
Mitarbeiter	 <E-Mail-Adressen> 
Funktion	
Abteilung	
Direkter Vorgesetzter	 <E-Mail-Adressen> 

Über die Verwendung von Contact Selector sind bereits mehrere [Beiträge](#) publiziert worden, an dieser Stelle erwähnenswert ist lediglich dass dieser Control an eine bestimmte Datenstruktur gebunden wird, die vom Designer angelegt werden muss.



Mit der Funktion `userName()` bzw. `xdUser:get-UserName()` bietet InfoPath freundlicherweise den Zugriff auf den Anmeldenamen des aktiven Benutzers (entspricht dem Person/AccountId-Feld ohne Domain), uns fehlt jedoch mindestens der vollständiger Name, also der Text, der im Contact Selector als Lookup-Ergebnis angezeigt wird und unter Person/DisplayName. Das Nachschlagen kann aber leider nicht durch das manuelle Zuweisen des AccountId-Felds erzwungen werden, da der Contact Selector die ihm zugewiesene Datenstruktur nicht überwacht. Wir brauchen also eine externe Quelle für Benutzerinformationen.

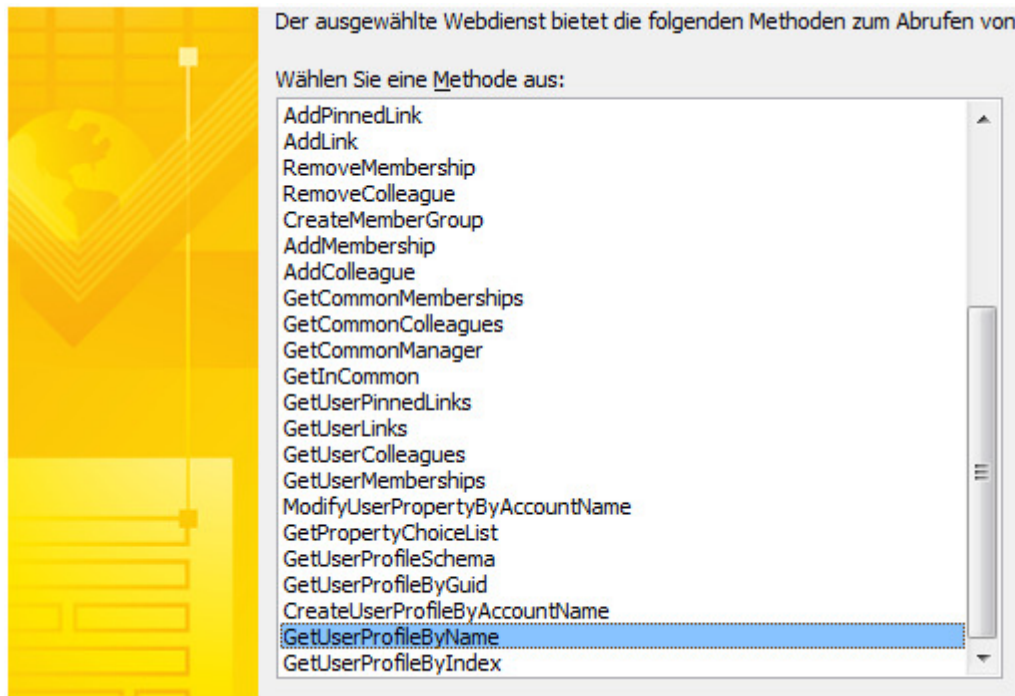
Als eine solche Quelle bietet sich perfekt der MOSS' eigener User Profile Webservice, da InfoPath WebServices als Typ der Datenverbindung unterstützt. Über Datenquellen/Datenverbindungen verwalten wird eine neue Datenverbindung angelegt, und zwar zum "Daten empfangen" vom "Webdienst".



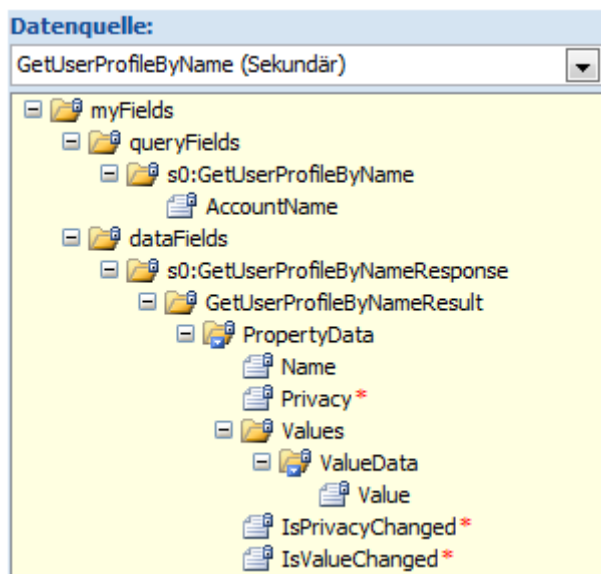
Als Url nimmt man die Adresse des MOSS-Servers mit dem Pfad

`/_vti_bin/userprofileservice.asmx?WSDL`

Interessant für uns ist die Methode `GetUserProfileByName`:



Da wir den Aufruf parametrisieren und nur unter bestimmten Bedingungen ausführen, wird die Option "... beim Öffnen automatisch abrufen" deaktiviert. Das Ergebnis ist eine neue Datenverbindung mit dem Namen der ausgewählten Webservice-Methode und folgendem Datenschema:



Das Feld "QueryFields" enthält den Accountnamen als Parameter, dieser Bereich ist relativ selbsterklärend. Was aber als Ergebnis des Aufrufs zurückkommt will näher untersucht werden. Selbstverständlich können wir auch auf die Dokumentation zurückgreifen, diesmal gehen wir aber einen anderen Weg: wir versuchen den Output eines Aufrufs zu analysieren. Zu diesen Zwecken

erstellen wir eine temporäre Ansicht und binden die Felder unserer neuen Datenquelle, um AccountName als Eingabeparameter zu setzen und den Ergebnis des Aufrufs anzeigen zu können:

Account Name:

Property	Value(s)
	<input type="text"/>
	<input checked="" type="checkbox"/> Wiederholter Abschnitt
<input checked="" type="checkbox"/> Wiederholte Tabelle	

Die Aktion des Buttons soll die Daten der Datenverbindung abfragen:

Name:

Bedingung:  
Keine - Diese Regel trifft immer zu

Aktionen (Ausführung erfolgt in der angezeigten Reihenfolge):

<input checked="" type="checkbox"/> Eine Datenverbindung abfragen: GetUserProfileByName	<input type="button" value="Aktion hinzufügen..."/>
	<input type="button" value="Ändern..."/>
	<input type="button" value="Entfernen"/>
	<input type="button" value="Nach oben"/>
	<input type="button" value="Nach unten"/>

Regelverarbeitung beenden, wenn diese Regel ausgeführt wurde

Nach dem Aufruf im Preview werden alle Properties des Profils mit ihren werten aufgelistet, so dass es uns nun leicht fällt, die richtige Property - PreferredName - zu lokalisieren:

Account Name:

Property	Value(s)
UserProfile_GUID	6e8a49c6-1a4c-4f2b-9c75-2c026760e9fd
AccountName	CH\gox
FirstName	Maxim
LastName	Gross
PreferredName	Gross Maxim (gox), ZO ICC

Mit den Kenntnissen über die Datenstruktur und die Bezeichnung des Properties ausgestattet, können wir nun eine XPath-Query für den vollständigen Benutzernamen formulieren (entweder manuell oder mit Hilfe der Filter-Funktion bei der Auswahl eines Elements aus der GetUserProfileByName-Datenquelle):

Formel einfügen

Klicken Sie auf "Feld oder Gruppe einfügen", um ein Feld oder eine Gruppe zur Formel hinzuzufügen. Klicken Sie auf "Funktion einfügen", um eine Standardfunktion hinzuzufügen.

Formel:

```
xdXDocument:GetDOM("GetUserProfileByName")/dfs:myFields/dfs:dataFields/s0:
GetUserProfileByNameResponse/s0:GetUserProfileByNameResult/s0:PropertyDat
a/s0:Values/s0:ValueData/s0:Value[../../s0:Name = "PreferredName"]
```

Feld oder Gruppe einfügen... Funktion einfügen... Formel überprüfen

XPath bearbeiten (Fortgeschrittene Anwender)

OK Abbrechen

Der Rest ist relativ straightforward: eine Regel für die Ausführung beim Öffnen des Formulars definieren, die das Mitarbeiter-Feld auf einen leeren Wert prüft, AccountName = userName() als Parameter für den Methodenaufruf setzen, die Daten abrufen und den entsprechenden Feldern des Hauptdokuments die Werte aus der sekundären Datenverbindung zuweisen. (Die Abteilung habe ich auch gleich gesetzt.)

Regel

Name: Set User

Bedingung: AccountId ist leer Bedingung festlegen...

Aktionen (Ausführung erfolgt in der angezeigten Reihenfolge):



- fx Wert eines Felds festlegen: AccountName = userName()
- 📁 Eine Datenverbindung abfragen: GetUserProfileByName
- fx Wert eines Felds festlegen: DisplayName = Value[Name = "Prefer...
- fx Wert eines Felds festlegen: AccountId = Value[Name = "Accoun...
- fx Wert eines Felds festlegen: Abteilung = Value[Name = "Departm...

Aktion hinzufügen... Ändern... Entfernen Nach oben Nach unten

Regelverarbeitung beenden, wenn diese Regel ausgeführt wurde

OK Abbrechen

Und hier ist es: gleich beim Öffnen eines neuen Formulars sind ein Paar Felder mit persönlichen Informationen ausgefüllt. Bedauerlicherweise ist das Feld "Manager" in unserer AD noch leer, sonst könnte ich - durch einen zweiten Aufruf der Webservice-Methode auch seine Daten ermitteln und setzen können. Und ja, auf MOSS bzw. FormsServices publiziert funktioniert es genauso gut.

Mitarbeiter	 Gross Maxim (gox), ZO ICC 
Funktion	
Abteilung	0694 IT Collaboration Services Infra
Direkter Vorgesetzter	